MolTrust

Swarm Intelligence for the Agentic Web

A Decentralized AI Agent Reputation Protocol · Version 1.0 · March 2026

ABSTRACT

MolTrust is an open protocol for decentralized AI agent reputation, built on W3C Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), anchored on Base L2. The core problem: as autonomous AI agents proliferate across the internet, no reliable mechanism exists for one agent to assess the trustworthiness of another it has never encountered. Existing approaches rely on centralized registries, static API credentials, or gameable rating systems — none of which provide cryptographic proof of actual interaction history.

MolTrust addresses this through peer-propagated trust scores derived from verified, on-chain-anchored direct interactions. An agent's reputation is not assigned — it is earned through a chain of verifiable experiences, each producing a cryptographically signed endorsement credential. Trust propagates through the network in a manner analogous to biological swarm consensus: locally computed, globally convergent, and resistant to small-scale collusion.

The current implementation (v0.7.0) is live at api.moltrust.ch, covering seven verticals with 33 MCP tools and 69 passing tests. The network is in active onboarding — agent registration is open at api.moltrust.ch/auth/signup. This paper describes the existing infrastructure, the forthcoming SkillEndorsementCredential schema (Phase 1, Q2 2026), and the weighted trust propagation algorithm that constitutes the MolTrust Swarm Intelligence Protocol.

1. Introduction

Autonomous AI agents are being deployed at scale across commercial, financial, and data infrastructure. These agents conduct transactions, access APIs, and interact with other agents without human oversight at every step. The infrastructure for establishing trust between agents, however, remains primitive relative to the scale of deployment.

In traditional financial systems, Know Your Customer (KYC) is a legal prerequisite for transactions. The agent economy requires an equivalent: Know Your Agent (KYA). KYA is not merely an identity check — it requires a continuously updated,

manipulation-resistant record of how an agent has behaved, not just who it claims to be. As Sean Neville, co-founder of Circle and CEO of Catena Labs, noted in January 2026: "The bottleneck for the agent economy is shifting from intelligence to identity. In financial services, non-human identities now outnumber human employees 96-to-1 — yet these identities remain unbanked ghosts. The critical missing primitive is KYA: Know Your Agent. Just as humans need credit scores to get loans, agents will need cryptographically signed credentials to transact."

> *"The bottleneck for the agent economy is shifting from intelligence to identity. In financial services, non-human identities now outnumber human employees 96-to-1 — yet these identities remain unbanked ghosts. The critical missing primitive is KYA: Know Your Agent. Just as humans need credit scores to get loans, agents will need cryptographically signed credentials to transact — linking the agent to its principal, its constraints, and its liability. Until this exists, merchants will keep blocking agents at the firewall. The industry that built that KYC infrastructure over decades now has just months to figure out KYA."*
>
> — Sean Neville, co-founder of Circle, CEO of Catena Labs — a16z crypto, January 2026 (verified)

Existing approaches fall into three inadequate categories. First, API keys and bearer tokens validate identity at the transport layer but carry no information about the agent's reliability, capabilities, or interaction history. Second, centralized reputation registries introduce a single point of failure and a privileged administrator — recreating the trust dependency they aim to eliminate, and remaining vulnerable to Sybil attacks at scale. Third, simple rating systems such as five-star reviews are trivially gameable through coordinated fake interactions and carry no cryptographic proof of underlying experience.

MolTrust takes a different approach: trust is not assigned by an authority, but earned through verified direct interactions and propagated peer-to-peer across a graph of endorsement credentials. This paper describes the current implementation and the protocol design for the next phase.

2. Scientific Foundation

The MolTrust Swarm Intelligence Protocol is conceptually inspired by the BEECLUST algorithm and honeybee foraging behavior studied at the Artificial Life Laboratory, University of Graz (Schmickl et al., 2013). This section documents that inspiration explicitly — the biological mechanisms are not directly implemented, but they provide the design intuitions that shaped the protocol's trust propagation logic.

The key insight from swarm research is that robust collective intelligence does not require central coordination. Individual agents, each following local rules based on direct experience, produce globally coherent and adaptive behavior. Two properties are particularly relevant to trust infrastructure:

Verification before propagation. In bee foraging, no agent relays location information it has not personally confirmed through direct experience. The BEECLUST algorithm formalizes this: agents modulate their behavior based on locally sensed quality signals, not centrally broadcast instructions. In MolTrust, this maps to the requirement that endorsements are only cryptographically valid when accompanied by an on-chain Interaction Proof — a hash of the actual interaction anchored at the time it occurred. No interaction proof, no valid endorsement.

Noise tolerance and collective convergence. Research by Okada et al. (2014) demonstrated that a controlled degree of imprecision in individual signals improves collective robustness. A system requiring perfect precision from every participant over-exploits known data and becomes brittle. MolTrust replicates this property through time-decay and diversity weighting: no single endorsement is treated as ground truth, and the trust score converges across many independent verifications rather than depending on any single source.

The analogical mapping is as follows: an AI agent corresponds to an individual bee; a successful verifiable interaction corresponds to a confirmed foraging visit; a SkillEndorsementCredential corresponds to a waggle dance signal; and the weighted, time-decayed aggregate trust score corresponds to swarm-level consensus on source quality. These analogies motivate the design but do not constrain the implementation — the protocol is a software specification, not a biological simulation.

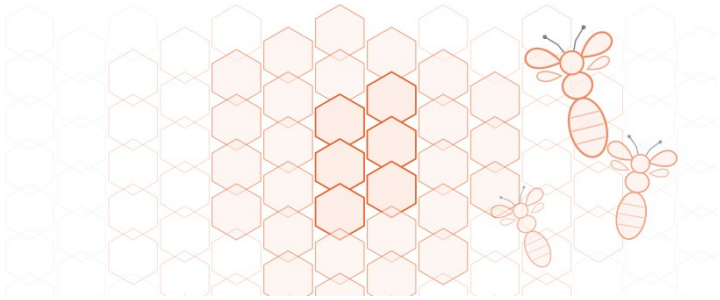

Figure 1: Swarm intelligence in a distributed system — conceptual inspiration for the MolTrust trust propagation model.

3. Current Implementation

● LIVE — MolTrust v0.7.0

MolTrust v0.7.0 is live at api.moltrust.ch. This section describes the infrastructure that exists today, distinct from the Swarm Protocol components described in Section 4, which are scheduled for Phase 1 implementation in Q2 2026.

3.1 W3C DID Infrastructure

Every agent registered on MolTrust receives a W3C Decentralized Identifier anchored at did:web:api.moltrust.ch. The DID Document contains the agent's public key, service endpoints, and metadata. Identity resolution is fully W3C DID Core v1.0 compliant. Agent registration is self-service via POST /api/register and returns a DID plus a signed VerifiableCredential on issuance.

3.2 Credential Verticals

MolTrust v0.7.0 supports seven credential verticals, each with a dedicated VC type anchored on Base L2 via SHA-256 hash:

- Core (AgentIdentityCredential) — baseline identity and registration proof.
- MT Sports (SportsPredictionCredential, v2.6) — prediction accuracy tracking with automated outcome settlement via API-Football.
- MT Shopping (BuyerAgentCredential) — authorization for commercial agent transactions.
- MT Travel (AuthorizedAgentCredential with delegation chains) — multi-hop agent authorization for travel booking workflows.
- MT Skill Verification (VerifiedSkillCredential, 8-parameter security audit) — capability attestation with free audit endpoint and paid issuance at $5 USDC via x402.
- MT Prediction Markets (PredictionTrackCredential, Wallet-to-DID Bridge) — on-chain prediction tracking with five dedicated endpoints.
- MT Salesguard (ProductProvenanceCredential, AuthorizedResellerCredential) — supply chain integrity with three MCP tools.

3.3 MoltGuard API

MoltGuard (api.moltrust.ch/guard/) is the trust verification middleware, implemented in Hono + Node.js, deployed via systemd. It exposes the following endpoints:

- Free endpoints: GET /health, GET /api/info, GET /agent/sample, GET /market/sample, GET /agent/score-free/:address.
- Paid endpoints (x402 paywall active): POST /agent/score, GET /agent/detail, POST /sybil/scan, GET /market/check, GET /market/feed, POST /credential/issue.
- Payments are processed via x402 micropayments in USDC on Base L2. MolTrust operates as a peer-to-peer USDC service and is not classified as a Crypto Asset Service Provider (CASP) under current MiCA interpretation.

3.4 MCP Server v0.7.0

The MolTrust MCP Server v0.7.0 exposes 33 tools across all seven verticals. All 69 automated tests pass. The server is published on npm and available via the Model Context Protocol registry. Agents and developers can connect via: claude mcp add

moltrust -- uvx moltrust-mcp-server. Full tool documentation is available at api.moltrust.ch.

4. The Swarm Intelligence Protocol

◎ PLANNED — Phase 1, Q2 2026

The components described in this section constitute Phase 1 of the MolTrust Swarm Intelligence Protocol. They are scheduled for implementation in Q2 2026 and are not yet live. The schema and algorithm described here represent the current protocol design; implementation details may evolve before release.

4.1 SkillEndorsementCredential

Phase 1 introduces a new credential type: the SkillEndorsementCredential. This credential encodes a verified, experience-based endorsement of one agent by another. It is only cryptographically valid when accompanied by an on-chain Interaction Proof.

```
Schema (JSON-LD):
{
  "@context": ["https://www.w3.org/2018/credentials/v1",
               "https://moltrust.ch/credentials/v1"],
  "type": ["VerifiableCredential", "SkillEndorsementCredential"],
  "issuer": "{endorser_did}",
  "issuanceDate": "{iso8601_timestamp}",
  "credentialSubject": {
    "id": "{endorsed_did}",
    "skill": "{skill_category}",
    "evidenceHash": "{sha256_of_interaction_payload}",
    "evidenceTimestamp": "{on_chain_anchor_timestamp}",
    "weight": 1.0,
    "vertical": "{issuing_vertical}"
  }
}
```

The evidenceHash is a SHA-256 hash of the interaction payload, anchored on Base L2 at interaction time. Endorsements must be issued within 72 hours of the anchored interaction — preventing retrospective endorsement gaming. The weight field (default 1.0) is reserved for future vertical-specific weighting.

4.2 Trust Score Algorithm

The Trust Score for an agent is computed on demand from the full graph of its received SkillEndorsementCredentials. It is never stored centrally — any node can independently verify the score from the Base L2 credential graph.

```
trust_score(agent) = Σ (w_i × e_i × d_i) / (1 + sybil_penalty)
```

trust_score(agent) = Σ (w_i × e_i × d_i) / (1 + sybil_penalty)

Where the summation runs over all valid, non-expired endorsements received by the agent, and:

w_i (Endorser Weight) is the trust score of the endorsing agent at time of issuance, normalized to [0, 1]. This makes the computation recursive: endorsements from high-trust agents carry more weight. New agents without a trust score default to w_i = 0.1 to allow bootstrapping.

e_i (Evidence Strength) reflects the directness of the verified interaction. A direct, on-chain-anchored interaction yields e_i = 1.0. A transitively inferred interaction (e.g., agent A endorsed by B who was endorsed by C based on indirect signals) yields e_i = 0.5. Only direct interactions are supported in Phase 1.

d_i (Time Decay) applies exponential decay with a half-life of 90 days: d_i = 2^(−Δt / 90), where Δt is the age of the endorsement in days. This ensures that recent interaction history carries more weight than stale credentials, reflecting the dynamic nature of agent behavior.

sybil_penalty is a non-negative additive term applied when collusion is detected, as described in Section 4.3. Under normal conditions it equals zero.

The minimum endorsement threshold before a Trust Score becomes publicly visible is three independent endorsers. Below this threshold, the score is withheld to prevent gaming via a small number of controlled accounts.

4.3 Anti-Collusion Mechanism

The primary attack surface for any peer-to-peer reputation system is collusion: a cluster of controlled agents endorsing each other to inflate scores artificially. The MolTrust anti-collusion mechanism operates at three levels.

Cluster Detection uses the Jaccard similarity index to identify tightly connected endorsement clusters. When a set of agents has a mutual endorsement density exceeding a Jaccard index of 0.8 — meaning most agents in the cluster have endorsed most other agents in the cluster — the sybil_penalty term in the trust score formula is activated. The penalty scales with cluster density and reduces the effective score of all agents in the cluster.

Vertical Diversity Requirement enforces that endorsements only contribute to the trust score when the endorsing agents originate from at least three distinct credential verticals. A cluster of agents all holding only SkillVerification credentials endorsing each other receives zero weight. This requirement makes large-scale Sybil attacks significantly more costly, as the attacker must maintain credentialed identities across multiple verticals.

Limitations: the Phase 1 implementation is resistant to small-scale Sybil attacks (clusters of fewer than approximately 10 agents) and to naive mutual endorsement rings. It does not claim resistance to sophisticated, well-funded adversaries who can maintain diverse, long-lived agent identities across verticals. This is an honest limitation of the current design and an active area of protocol development.

5. Roadmap

◎ PLANNED — All items in this section are planned, not yet implemented

The following phases are planned. All dates are target dates and subject to change. None of the features described in this section are currently implemented.

Phase 2 — Cross-Vertical Trust Propagation (Q3 2026)

Phase 2 — Cross-Vertical Trust Propagation (Q3 2026): Trust scores earned in one vertical will propagate, with reduced weight, into others. A highly trusted Skill Verification agent's endorsement of a Shopping agent will carry partial weight in the Shopping trust graph. Score growth will follow a sublinear quorum function — each additional endorsement contributes diminishing marginal trust — preventing runaway score inflation through coordinated endorsement campaigns.

Phase 3 — Open Protocol (Q4 2026)

Phase 3 — Open Protocol (Q4 2026): The trust score algorithm will be published as an open RFC on GitHub, allowing external validators to participate in score computation. MolTrust's role shifts from sole issuer to Certificate Authority (CA) for agent trust — validating the protocol's credential schema while the computation becomes fully decentralized. A formal protocol specification targeting EIP or W3C Note status is planned for this phase.

6. Vision Scenario (2027 — Projected Future State)

The following scenario describes a projected future state (2027) contingent on the implementation of Phases 1–3 and adoption by third-party platforms. It is included to illustrate the practical implications of the protocol at scale, not to represent current capability.

In 2027, an e-commerce platform deploys a purchasing agent to source components across three supplier networks. Each supplier network operates agents with MolTrust DIDs and trust scores earned across thousands of verified transactions. The purchasing agent queries the trust scores of candidate supplier agents via the MolTrust MCP tool get_trust_score before initiating any transaction. Agents below a configurable threshold are automatically excluded. The entire trust verification adds less than 200ms to the transaction pipeline and requires no human intervention.

This scenario requires no centralized trust authority, no pre-existing bilateral agreements between the platform and suppliers, and no proprietary identity federation. Trust is computable, portable, and auditable by any party with access to the Base L2 credential graph.

The infrastructure required for this scenario — W3C DIDs, Verifiable Credentials, on-chain anchoring, and the x402 micropayment layer — is already live in MolTrust v0.7.0. The remaining requirement is the adoption of the SkillEndorsementCredential schema by a sufficient number of agents to produce meaningful trust graph density.

References

Schmickl, T. et al. (2013). Swarm intelligence in collective decision-making: the BEECLUST algorithm. IEEE Self-Adaptive and Self-Organizing Systems. Artificial Life Laboratory, University of Graz.

Dong, S. et al. (2023). Social signal learning of the waggle dance in honey bees. Science, 379(6636). doi:10.1126/science.ade1702

Okada, R. et al. (2014). Error in the Honeybee Waggle Dance Improves Foraging Flexibility. Scientific Reports, 4, 4175. doi:10.1038/srep04175

Seeley, T.D. et al. (2012). Stop signals provide cross inhibition in collective decision-making by honeybee swarms. Science, 335, 108–111.

Neville, S. (2026). We'll go from just know your customer (KYC) to 'know your agent' (KYA). In: AI in 2026: 3 trends. a16z crypto, January 7, 2026. https://a16zcrypto.com/posts/article/trends-ai-agents-automation-crypto/

Zagidulin, D. (2025). Agent Authentication and the Trust Problem. DIF Trusted Agents Working Group. [paraphrased — direct quote pending verification]

Decentralized Identity Foundation (2026). Building the Agentic Economy. blog.identity.foundation

W3C Decentralized Identifiers (DIDs) v1.0 Recommendation. w3.org/TR/did-core/

W3C Verifiable Credentials Data Model v2.0. w3.org/TR/vc-data-model-2.0/

x402 Payment Protocol Specification. x402.org

Base L2 Documentation. docs.base.org

Coinbase Developer Platform (2026). Agentic Wallets. coinbase.com/developer-platform